WHAT IS CLAIMED IS:

1. A data structure comprising:

a thread control block, wherein said thread control block is described by a first data structure; and

a message, wherein said message is described by a second data structure and said first data structure comprises said second data structure.

2. The data structure of claim 1, wherein said first data structure is configured to store information used to control execution of a thread.

Sub B1

3. The data structure of claim 1, wherein said second data structure is configured to store a message.

4. The data structure of claim 1, wherein said first data structure further comprises:

a process control block pointer, wherein said process control block pointer points to a process control block;

processor information; and

stack information.

5. The data structure of claim 4, wherein said process control block comprises:

memory information;

thread information;

device driver information; and

stack information.

6. The data structure of claim 4, wherein said processor information comprises:

a processor identifier; and

thread information;.

7. The data structure of claim 1, wherein said second data structure further comprises:

control information.

8. The data structure of claim 7, wherein said second data structure further comprises:

data.

9. An operating system, wherein said operating system is configured to provide a user space and a kernel space, comprising:

a plurality of tasks, wherein said tasks are executed in said user space;

a thread control block/message structure, wherein said thread control block/message structure comprises

a thread control block, wherein said thread control block is described by a first data structure, and

a message, wherein said message is described by a second data structure and said first data structure comprises said second data structure; and

a microkernel wherein

said microkernel is executed in said kernel space, and

said microkernel is configured to support inter-task communication by virtue of being configured to pass said thread control block/message structure from a first one of said tasks to a second one of said tasks.

10. The operating system of claim 9, wherein said first data structure is configured to store information used to control execution of a thread.

11. The operating system of claim 9, wherein said second data structure is configured to store a message.

12. The operating system of claim 9, wherein said first data structure further comprises:

a process control block pointer, wherein said process control block pointer points to a process control block;

processor information; and

stack information.

13. The operating system of claim 12, wherein said process control block comprises:

memory information;

thread information;

device driver information; and

stack information.

14. The operating system of claim 12, wherein said processor information comprises:

a processor identifier; and

thread information;.

15. The operating system of claim 9, wherein said second data structure further comprises:

control information.

16. The operating system of claim 15, wherein said second data structure further comprises:

data.

17. The operating system of claim 9, wherein said operating system employs a client/server architecture.

18. The operating system of claim 17, wherein said first task acts as a client task and said second task acts as a server task.

19. A method of inter-task communication comprising:

sending a message between a first task and a second task by performing a send operation, wherein said first task performs said send operation and said send operation employs a thread control block/message structure; and

causing said second task to perform a receive operation.

20. The method of claim 19, wherein said thread control block/message structure comprises:

a thread control block, wherein said thread control block is described by a first data structure, and

a message, wherein said message is described by a second data structure and said first data structure comprises said second data structure.

22. The method of claim 19, wherein said thread control block/message structure supports control of a thread within said second task and said method further comprises:

determining if said thread is queued to a thread queue of said second task; and

transferring said message from said first task and said second task.

23. The method of claim 21, wherein said transferring said message comprises:

passing said message between said first task and said second task by performing a fast-path message copy if said thread is queued to said thread queue; and

passing said message between said first task and said second task by performing a message copy if said thread is not queued to said thread queue.

23. The method of claim 22, wherein said performing said fast-path message copy comprises:

copying said message from a memory space of said first task to a memory space of said second task.

24. The method of claim 22, wherein said performing said message copy comprises:

copying said message from said first task to said thread control block/message structure;

waiting for said thread to be queued to said thread queue; and

copying said message from said thread control block/message structure to said second task.

25. The method of claim 22, wherein said first task acts as a client task and said second task acts as a server task.

26. A computer program product encoded in computer readable media, said computer program product comprising:

a first set of instructions, executable on a computer system, configured to send a message between a first task and a second task by performing a send operation, wherein said first task performs said send operation and said send operation employs a thread control block/message structure;

a second set of instructions, executable on said computer system, configured to cause said second task to perform a receive operation.

27. The computer program product of claim 26, wherein said thread control block/message structure comprises:

a thread control block, wherein said thread control block is described by a first data structure, and

a message, wherein said message is described by a second data structure and said first data structure comprises said second data structure.

28. The computer program product of claim 26, wherein said thread control block/message structure supports control of a thread within said second task and said computer program product further comprises:

a third set of instructions, executable on said computer system, configured to determine if said thread is queued to a thread queue of said second task; and

a fourth set of instructions, executable on said computer system, configured to transfer said message from said first task and said second task.

29. The computer program product of claim 28, wherein said fourth set of instructions comprises:

a first subset of instructions, executable on said computer system, configured to pass said message between said first task and said second task by performing a fast-path message copy if said thread is queued to said thread queue; and

a second subset of instructions, executable on said computer system, configured to pass said message between said first task and said second task by performing a message copy if said thread is not queued to said thread queue.

30. The computer program product of claim 29, wherein said first subset of instructions comprises:

a first sub-subset of instructions, executable on said computer system, configured to copy said message from a memory space of said first task to a memory space of said second task.

31. The computer program product of claim 29, wherein said second subset of instructions comprises:

a first sub-subset of instructions, executable on said computer system, configured to copy said message from said first task to said thread control block/message structure;

a second sub-subset of instructions, executable on said computer system, configured to wait for said thread to be queued to said thread queue; and

a third sub-subset of instructions, executable on said computer system, configured to copy said message from said thread control block/message structure to said second task.

32. The computer program product of claim 29, wherein said first task acts as a client task and said second task acts as a server task.

33. A computer system comprising:

a processor;

computer readable medium coupled to said processor; and

computer code, encoded in said computer readable medium, configured to cause said processor to:

send a message between a first task and a second task by performing a send operation, wherein said first task performs said send operation and said send operation employs a thread control block/message structure; and

cause said second task to perform a receive operation.

34. The computer system of claim 33, wherein said thread control block/message structure comprises:

a thread control block, wherein said thread control block is described by a first data structure, and

a message, wherein said message is described by a second data structure and said first data structure comprises said second data structure.

35. The computer system of claim 33, wherein said thread control block/message structure supports control of a thread within said second task and said computer code is further configured to cause said processor to:

determine if said thread is queued to a thread queue of said second task; and

transfer said message from said first task and said second task.

36. The computer system of claim 35, wherein said computer code further configured to cause said processor to transfer said message from said first task and said second task is further configured to cause said processor to:

pass said message between said first task and said second task by performing a fast-path message copy if said thread is queued to said thread queue; and

pass said message between said first task and said second task by performing a message copy if said thread is not queued to said thread queue.

37. The computer system of claim 36, wherein said computer code further configured to pass said message between said first task and said second task by performing a fast-path message copy is further configured to cause said processor to:

copy said message from a memory space of said first task to a memory space of said second task.

38. The computer system of claim 36, wherein said computer code further configured to pass said message between said first task and said second task by performing a message copy is further configured to cause said processor to:

copy said message from said first task to said thread control block/message structure;

wait for said thread to be queued to said thread queue; and

copy said message from said thread control block/message structure to said second task.

39. The computer system of claim 36, wherein said first task acts as a client task and said second task acts as a server task.

add 91